# Ground-Truthing AI Energy Consumption: Validating CodeCarbon Against External Measurements

Raphael Fischer Lamarr Institute, TU Dortmund University *raphael.fischer@tu-dortmund.de*

*Although machine learning (ML) and artificial intelligence (AI) present fascinating opportunities for innovation, their rapid development is also significantly impacting our environment. In response to growing resource-awareness in the field, quantification tools such as the* ML Emissions Calculator *and* CodeCarbon *were developed to estimate the energy consumption and carbon emissions of running AI models. They are easy to incorporate into AI projects, however also make pragmatic assumptions and neglect important factors, raising the question of estimation accuracy. This study systematically evaluates the reliability of static and dynamic energy estimation approaches through comparisons with ground-truth measurements across hundreds of AI experiments. Based on the proposed validation framework, investigative insights into AI energy demand and estimation inaccuracies are provided. While generally following the patterns of AI energy consumption, the established estimation approaches are shown to consistently make errors of up to 40%. By providing empirical evidence on energy estimation quality and errors, this study establishes transparency and validates widely used tools for sustainable AI development. It moreover formulates guidelines for improving the state-of-the-art and offers code for extending the validation to other domains and tools, thus making important contributions to resource-aware ML and AI sustainability research.*

*Keywords: Energy, Resource-Aware Machine Learning, Sustainable Artificial Intelligence, Sustainability, CodeCarbon*

## 1 Introduction

Artificial intelligence (AI) holds many promises for our world, however does not come without costs. While modern AI enables sustainable innovation in domains such as construction, transportation, healthcare, and manufacturing [31], it also poses critical risks to our society, economy, and environment—or, in other words, across all sustainability dimensions [59, 14]. The particular danger of "systemic risks" and resulting need for "environmental sustainability" was manifested by experts worldwide, for example in the *International AI Safety Report* [3]. Unfortunately, the field of machine learning (ML) lacks resource-awareness and is mostly focused on boosting predictive performance [15]. Given that publications often fail to justify their compute expenses or discuss potential negative consequences [5], the observable "compute trends" [47] and "bigger-is-better paradigm" [57] are concerning but not surprising.

In order to use and advance AI in a sustainable manner, it is therefore imperative to balance predictive capabilities with resource consumption [18, 16], or put differently, acknowledge and navigate the "Pareto front" of "multi-dimensional model performance" [14, 57]. Various works explored such trade-offs by investigating the energy consumption and carbon emissions of using AI for generative purposes [36] and classic learning tasks, such as computer vision [19, 46, 17] or natural language processing [50, 37]. Because respective studies require "systematic and accurate measurements" of compute demand [24], the community developed tools for quantifying the energy consumption of ML. Among the first projects is the *ML Emissions Calculator* [38], which statically estimates energy consumption and resulting $CO_2$-equivalents based on user-provided information about the performed experiments. Shortly later, the *experiment-impact-tracker* was published as a straightforward drop-in library for dynamically tracking compute utilization and estimating carbon emissions of running Python code [24]. Since the tool was not developed much further,

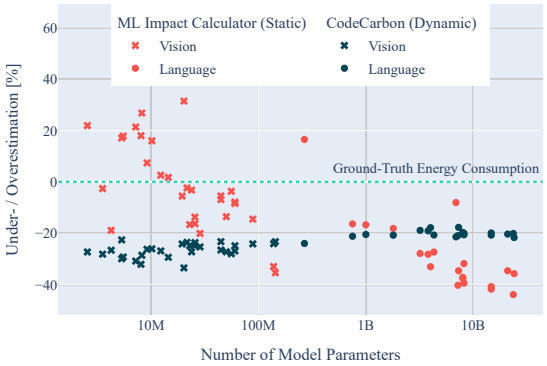


Figure 1: Compared to ground-truth energy measurements, the *ML Emissions Calculator* [38] and *CodeCarbon* [10] often under- or overestimate the resource demand of AI.

it was quickly outshined by *CodeCarbon* [10], a library that does not only allow for integrating resource estimations into Python code, but moreover offers an interactive dashboard view and useful emission comparisons (e.g., household consumption or car driving). All tools became widely adopted by resource-aware practitioners, with the *ML Emissions Calculator* being cited over 1000 times on *Google Scholar* [38] and *CodeCarbon* surpassing 1500 stars on *GitHub* and nearly 2 million downloads on *PyPI* [10].

While the aforementioned initiatives are without a doubt valuable contributions for sustainable AI development, a crucial question remains: **How accurately do they estimate the energy consumption of AI?** Static approaches like the *ML Emissions Calculator* assume a constant power draw of the hardware, which however might change across different experiments. Dynamic estimations via *CodeCarbon* consider the actual compute utilization, however neglect hardware components that cannot be profiled from software, such as the power supply unit, cooling overhead, or peripheral devices. Figure 1 summarizes these issues as a teaser to later investigations, with each point representing a single AI experiment. In comparison with ground-truth measurements (turquoise), both the static (salmon) and dynamic (teal) tools under- or overestimate the energy consumption of most experiments by up to 40%. This motivates the study at hand, which sheds light on the unknown figures of AI resource consumption via four key contributions:

- Methodology for validating static and dynamic resource estimation approaches (see Figure 2)
- Code base for reproducing and extending the experiments to other domains and estimation approaches
- Experimental results that explore AI resource consumption and estimation errors
- Discussion on related works, takeaways, limitations, and implications for future research

By investigating the ground-truth energy demand of 50 models from the vision and language domain, this work fills an important gap in sustainable AI literature. It provides empiric evidence that popular tools like *CodeCarbon* and the *ML Emissions Calculator* should be enhanced to account for the missing factors in their internal estimation procedures. These insights empower researchers and practitioners to make informed decisions about the tools and methods they use for sustainability assessments, carbon accounting, and energy-aware AI system design and deployment.

# 2 Methodology

The core concepts for validating the estimation of AI energy consumption are schematically summarized in Figure 2, guiding this section. Practitioners who want to assess the environmental impacts of using or developing AI generally have three ways for quantifying respective numbers, to be introduced next. The terminology was aligned with the framework for sustainable and trustworthy reporting [14, 15]—AI *experiments* (i.e., evaluations) are *executed* in an *environment*, which represents the hardware and software. The experiment is characterized by a *configuration*, which describes the learning task, dataset, and AI model (or method) at hand. While this work focuses on AI deployment (i.e., inference with pre-trained models), the following can be easily applied to other tasks.

## 2.1 Static Estimation

At the most abstract level, the configuration and environment of any executed experiment can be utilized to statically estimate the corresponding energy demand and resulting environmental impact. This approach is also implemented by the *ML Impact Calculator* [38], which estimates the impact via the total amount of $CO_2$-equivalents:

$$\texttt{CO}_2\texttt{-Equiv} = \texttt{Power} \cdot \texttt{Time} \cdot \texttt{CO}_2\ \texttt{Efficiency} \quad (1)$$

All of the factors are constants derived from the given configuration and environment, with the power consumption in Watt commonly reflecting the main processor's thermal design power (TDP) or processor base power (PBP), resulting in $\texttt{Energy} = \texttt{Power} \cdot \texttt{Time}$. This pragmatic simplification understands the main processor as the biggest factor for the overall energy demand of running AI, which is in line with respective studies [25], however hardly reflects the complexity of modern execution environments. The $\texttt{CO}_2\ \texttt{Efficiency}$ describes the amount of $CO_2$ emitted per unit energy, which is primarily subject to the local energy mix [38]. Static estimations only require to measure the running time and are thus easiest to perform, however fail at accounting for dynamic compute utilization and hardware

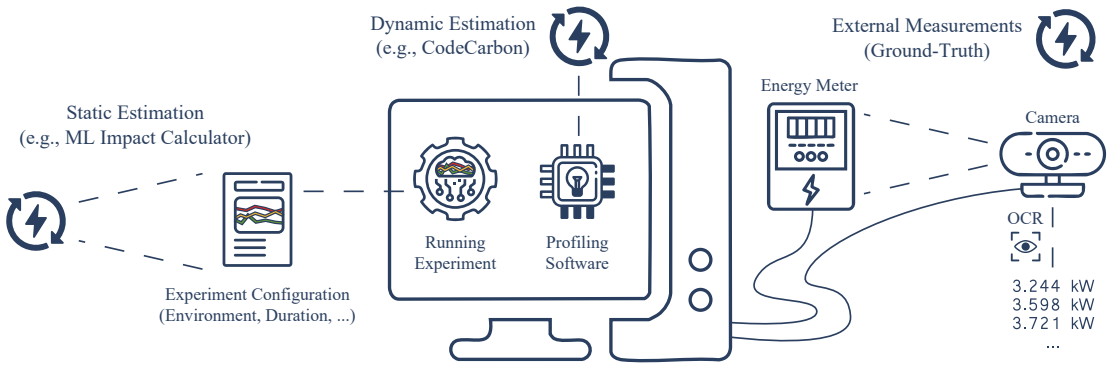


Figure 2: Schematic visualization for validating the static and dynamic estimation of energy demand via external measurements.

complexity. To give some examples for non-captured phenomena, running models of varying size will affect the power draw, performing different tasks (e.g., training or inference) might result in differences of resource demand, and compute utilization of a single experiment could even change over time. In short, while serving practitioners with a straightforward solution to report on the environmental impacts of running AI, static estimations remain rather vague approximations.

## 2.2 Dynamic Estimation

The second estimation approach aims at accounting for dynamic resource demand, thus making the $\texttt{Power} \times \texttt{Running Time}$ in Equation 1 less static. In practice, this requires to run additional *profiling software* on the execution environment at hand, such as the aforementioned *experiment-impact-tracker* [24] or *CodeCarbon* [10]. Internally, these libraries profile the power consumption of the central or graphics processing unit (CPU / GPU) via manufacturer tools such as *Intel*'s running average power limit [29] and *NVIDIA*'s management library [41]. They allow to iteratively capture the current processor power draw at any point in time ($t$). By performing respective measurements in short temporal intervals ($\Delta_{\texttt{Time}} = t - (t-1)$), it is possible to estimate the resulting energy draw over time, i.e., $\texttt{Energy} = \sum_{t=1}^{T} \texttt{Power}_t \cdot \Delta_{\texttt{Time}}$.

Unfortunately, few hardware systems have built-in capabilities for tracking the power consumption of other components than processors, neglecting for example the resource demand of peripheral devices, power supply, and cooling. The overhead impact of the latter recently received more attention, with studies revealing that cooling can attribute to over 10% of the overall AI power consumption [64, 32]. Moreover, special adaptions are required for profiling specialized hardware devices, such as edge accelerators by *Intel* and *Google* [49], or embedded computing boards (e.g, *NVIDIA Jetson*) [18]. It should also be noted that internal hardware profiling usually requires administrative access rights to the system and can cause a marginal energy draw overhead. To summarize, dynamic estimation solutions are easy to incorporate and likely better capture the actual resource consumption of the system, however face limitations with regard to supported hardware and underestimation.

## 2.3 Validation via External Measurements

Overcoming the aforementioned limitations requires to somehow measure the ground-truth $\texttt{Energy}$ consumption. In order to capture the impact of all environment components, such measurements ideally need to be taken from outside of the system (i.e., externally). The ground-truth measurements discussed in this work were obtained from a simple yet effective approach, using a standard energy meter, a basic camera, and some straightforward computer vision code. This setup is easy to realize and can thus be translated to many AI experiment scenarios.

Energy meters exist in various forms, from traditional analogue devices to smart gadgets [4]. Basic variants are not only highly affordable (less than 10€), but can also be easily installed between the power socket and plug of the hardware on which the AI experiment is executed. While such simple meters do not usually allow for digitally extracting the measurements, the display can be easily tracked with a standard camera. This setup allows to capture images of the displayed energy consumption at the start and end of any experiment, which can later be processed to numeric data via basic computer vision and optical character recognition (OCR).

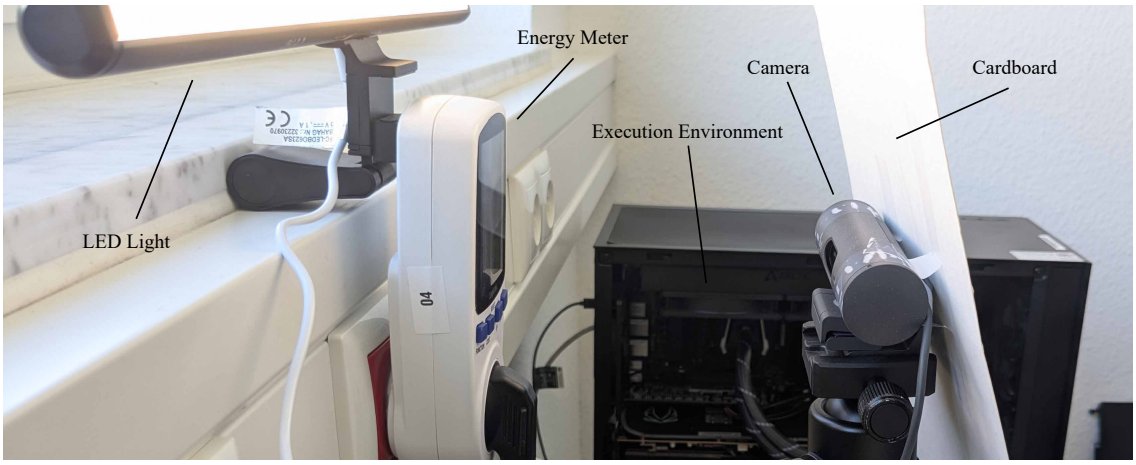


Figure 3: Hardware setup for validating energy estimation with ground-truth data from an energy meter, tracked with a camera.

Figure 2 visualizes how all three approaches can be run in parallel, using only one execution environment equipped with some basic additional devices. By storing and aggregating the respective logs, it is possible to compare and validate static estimation, dynamic estimation, and external measurements of AI energy consumption. It is important to note that external measurements also have certain drawbacks—most importantly, they require additional hardware and implementation effort. In fact, all three approaches have their pros and cons, with this work exploring their relation and complementary nature (see also the later Discussion).

## 2.4 Practical Setup

To empirically compare the different approaches for quantifying AI energy consumption, the introduced concepts were also practically implemented. Figure 3 showcases the experimental setup, using a basic energy meter by *LogiLink*, a camera by *Logitech*, and a deep learning workstation equipped with an *Intel i9-13900K* CPU and *NVIDIA RTX 4090* GPU. To allow for overnight experiments and alleviate display reflections, the setup also uses an LED light and a white piece of cardboard. The experiment software was implemented in Python and can be found at `https://github.com/raphischer/ai-energy-validation`, which also includes the logs and results. The dynamic estimate was performed with *CodeCarbon 3.0.1* [10], while the static estimate assumed a TDP of 300 Watt for the GPU (taken from the *ML Impact Calculator* [38]) and 125 Watt for the CPU (constant in *CodeCarbon*'s `cpu_power.csv` file). As additional dependencies, experiment execution was streamlined with *mlflow* [61], camera access and computer vision was enabled by *opencv-python* [6], OCR was performed with *scikit-learn* [43] (using a custom random forest classifier), the logs were analyzed with *pandas* [56], and all plots were created with *plotly* [44].

Because AI deployment quickly exceeds the energy demand of training, the experiments focused on the inference performance of pre-trained AI models. To validate AI energy consumption across different configurations, the performance of 30 image classifiers [48, 52, 22, 26, 23, 28, 9, 45, 65, 27, 53, 51, 54, 34], available from *Keras 3* [8], as well as 20 large language models [30, 63, 21, 2, 1, 55, 11, 20, 40, 60, 42] offered by *Ollama 0.11.8* [7] was measured. Later referred to as *Vision* and *Language*, these open weights models represent examples from two popular deep learning application domains. All models were published between 2015 and 2025 and have a complexity between 2 million and 23 billion parameters, making them feasible for single-GPU deployment (i.e., on-premise AI instead of AI-as-a-service [33]). To assess their performance, the vision models were configured to perform image classification on *ImageNet* samples [12] for 2 minutes, testing inference on the CPU or GPU with a batch size of 4 or 16 (i.e., processing 4 or 16 images at a time). The language models were confronted with random prompts from the *Puffin* dataset [39] for 15 minutes, using a temperature of 0.1 or 0.7 (lower value makes the models behave more deterministic with regard to token probabilities). If not otherwise specified, the following displays results for the default batch size of 16 and temperature of 0.7. All results were averaged over three consecutive runs that reduce random outlying behavior, with standard deviation information indicated by error bars.

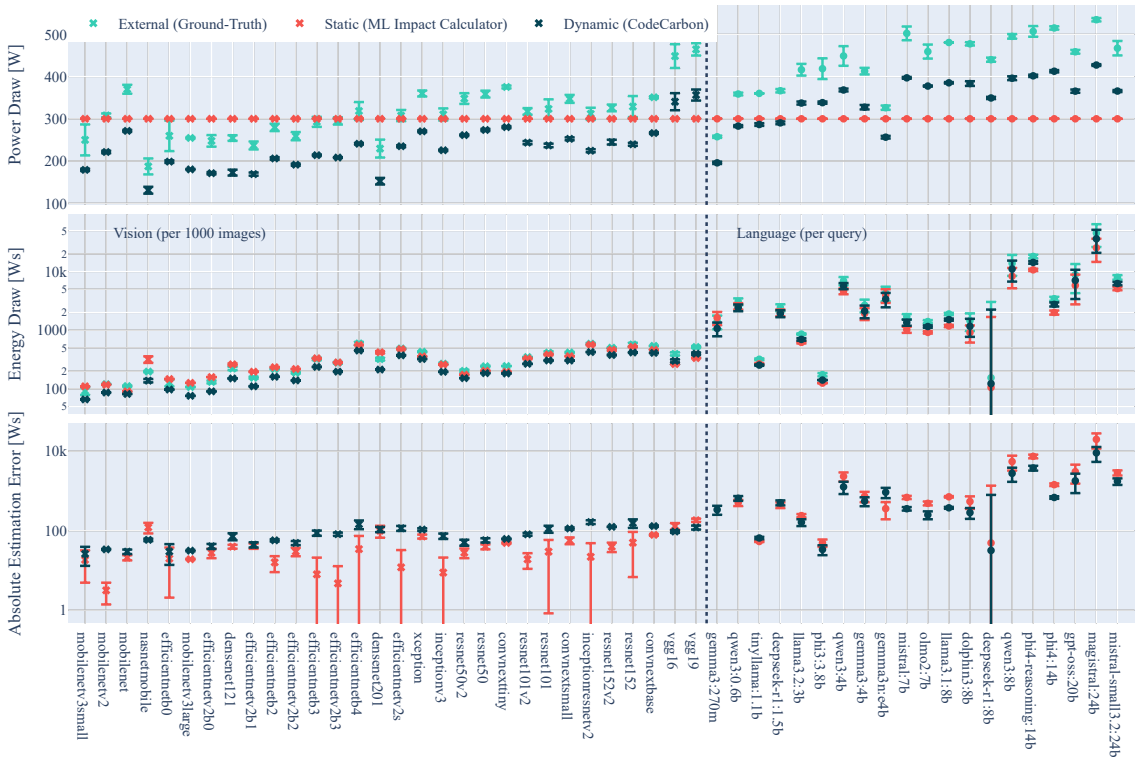


Figure 4: Power draw of the execution environment (first row), energy draw for performing inference with a fixed number of samples (second row), and energy estimation errors (third row), assessed for all three approaches and across all tested models. There is a clear upwards trend of power, energy, and error, generally growing with model complexity. The energy estimation approaches follow the general patterns of the ground-truth data but make significant estimation errors.

Running all experiment runs took less than three days and resulted in carbon emissions based on the consumed energy, connecting back to quantifying environmental impacts [38] via Equation 1. With a ground-truth energy consumption of 20.04 $\texttt{kWh}$, and assuming a $\texttt{CO}_2\ \texttt{Efficiency}$ of 0.38 for Germany [62], the total emissions of the experiments are estimated to $20.04\ \texttt{kWh} \cdot 0.38\ \texttt{kgCO}_2\texttt{/kWh} \approx 7.62\ \texttt{kg CO}_2\texttt{-Equiv}$ (not taking into account any embodied factors [13] or overhead efforts from development, testing, and writing). Keep also in mind this work only focuses on model performance in terms of resource consumption and does not explore any efficiency trade-offs with prediction quality [14].

## 3 Results

Recall that a high-level summary of the experimental results was already given in Figure 1, however the following explores them in more depth. To investigate the alignment of energy estimations and ground-truth data, the first row of Figure 4 displays the average power draw (Watt, $y$-axis) of all tested models ($x$-axis) during the experiment. The models were sorted based on their parameter count, from *mobilenetV3small* (2.5m parameters) [27] to *mistral-small3.2:24b* [40], resulting in a visible upwards trend of power consumption due to higher compute utilization. Nevertheless, there is clear evidence that the power draw does not only depend on the parameter count—some of the models are more power-efficient than expected from their complexity, such as *nasnetmobile* [65], *densenet201* [28], and the *gemma3* variants [20], Looking at the different curves, we also see that the constant value assumed by the static energy estimation (salmon points at 300 Watt) differs from the ground-truth measurements (turquoise), especially for the smallest and largest models. The dynamic profiling (teal) follows the general upwards trend, however compared to the externally measured data, consistently underestimates the power draw. The absolute difference between the measurements and estimates often exceeds 100 Watt, which is a significant error considering the TDP of 300 Watt [38], Impressively, the maximum observed power draw (534 Watt for *magistral:24b* [40]) is nearly twice as high as the static estimate.

In the second row of Figure 4, we see the amount of energy (Watt-seconds, logarithmic $y$-axis) required to classify 1000 images (vision) or answering a single query (language). Once again showcasing an upwards trend, this demonstrates the considerably higher resource demand of generative language models compared to image classifiers [36]. While all investigated models belong to the family of deep neural networks and consist of many million parameters, the classifiers are clearly faster and more energy-friendly, processing many thousand images in the time that language models need for answering a single prompt. As before, the energy demand does not strictly increase with parameter count—some models like *tinyllama:1.1b* [63] and *phi3:3.8b* [1] answer prompts with much less energy than expected from the complexity. We moreover see that both estimation approaches are generally able to follow the rough patterns of energy consumption, however also make errors. The static estimates are too high for small models and too low for bigger models, while the dynamic approach always underestimates the energy demand. One should note that these differences are expected to increase when running the models for longer periods of time, as the constant mismatch of power consumption (first row) will increase the gap over time.

The absolute energy estimation errors are displayed in the last row of Figure 4, connecting back to the relative results in Figure 1. The absolute values were used in order to also display the errors on a logarithmic scale. For smaller classifiers, the static estimation results in lower errors (note that some of them actually represent an overestimation), while for the bigger models, the dynamic estimates are more accurate. This can also be observed in the other rows, where the teal (dynamic) data "overtakes" the salmon (static) estimations with growing model complexity. As we once again find an upwards trend in the bottom row, we also see evidence that the estimation error tends to grow with model size and resource demand. Estimation errors below 10 Watt-seconds are extremely rare and could only be observed for four of the vision models. Across all three rows of the plot, the standard deviations across the experimental runs are generally rather low, demonstrating stability of the results with respect to randomization effects—except for *deepseek-r1:8b* [11], for which strong variations across different queries are observed.

As another point of this experimental investigation, we next explore how the batch size and temperature configuration impacts the resource consumption of vision and language models, respectively. Figure 5 showcases the ground-truth energy demand (top row) as well as the energy estimation errors (bottom row) when configuring the models with these hyperparameters and once more classifying 1000 images or answering a single query—the resulting performance distribution is summarized via box plots. One should note that the image batch size choice (left) only affects the running time and resource demand of the vision models, however does not impact their prediction quality. The results demonstrate that larger batches result in lower energy draw and thus also reduce the static and dynamic estimation errors across all models, however the batch size is naturally capped by the GPU memory capacity (i.e., larger batch sizes would not have been feasible for some of the more complex models). The temperature (right) actually affects the language model output, as it weights the predicted token probabilities to control how explorative or deterministic the provided query is answered. We see that the lower temperature (i.e., higher focus on token probabilities) leads to higher energy demand and larger estimation errors across the models. However, this configuration also seems to introduce more variation, resulting in considerably larger boxes. These results evidence that hyperparameters can have a strong impact on the resource demand of AI models, necessitating to explicitly report on the specific configuration of evaluated models.

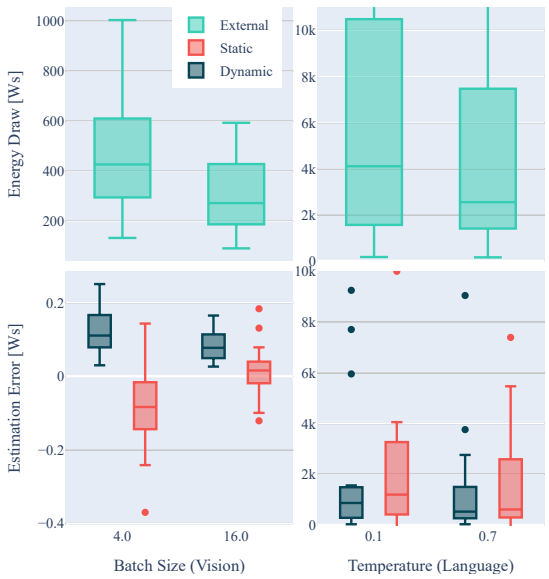


Figure 5: Impact of hyperparameter choice ($x$-axes) for AI energy demand. The higher batch size and temperature results in more efficient compute utilization (i.e., lower energy draw) for vision and language models, which also lead to lower estimation errors.

While deep learning models have become especially pop-

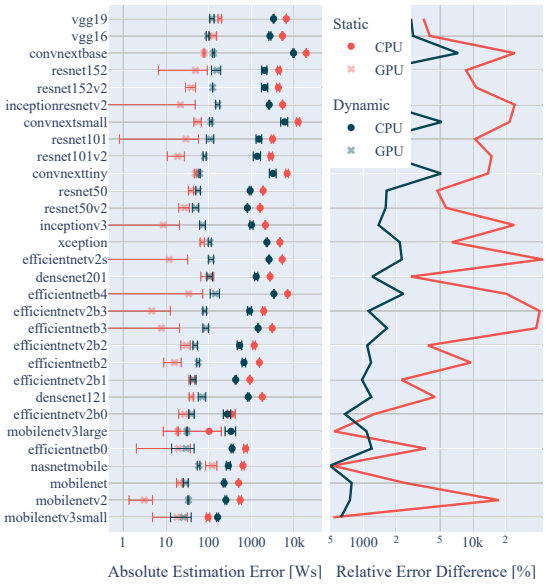


Figure 6: Absolute energy estimation errors of vision models when deployed on the CPU or GPU (left). Both the static and dynamic approaches make higher errors for CPU inference, and the differences between the CPU and GPU errors (right) grow with model complexity.

ular thanks to the efficiency of GPU deployment, there are also cases where practitioners are restricted to running AI on a CPU. This raises the question of whether energy estimation inaccuracies are affected by performing inference on the CPU or GPU. For this investigation, Figure 6 lists the respective absolute estimation errors (left logarithmic $x$-axis) for the vision models, when using or disabling the GPU—accordingly, the faded scatter points represent the same data displayed in the last row of Figure 4. This comparison clearly shows that the errors of both estimation approaches are considerably higher when only utilizing the CPU. As before, the errors grow with model size (i.e., upwards along the $y$-axis), and the differences between the CPU and GPU errors seem to also increase. In the most extreme case (*EfficientNetV2s* [54]), the dynamic estimation error for CPU inference (5225 Ws) is over 463 times as high as the respective GPU error (12 Ws). On the right hand side, the difference between the CPU and GPU estimation errors is shown more explicitly, in relation to the GPU errors. It demonstrates that the relative CPU error also increases with model complexity and is more than ten times (1000%) higher for nearly all configurations. As before, variations can be observed across the models—the *ConvNeXt* [34] and *EfficientNet* [53] variants have the highest error differences, while some models with comparable parameter count show less deviations. Moreover, the static estimation error differences (using a TDP of 125 Watt for the CPU) are considerably higher than the ones observed for dynamic estimations (except for *MobileNetV3Large* [27]). As such, the results indicate that the choice of execution environment and processor strongly impacts the accuracy of energy estimation tools, expecting higher errors for CPU-only inference.

# 4 Discussion

Let us interpret the experimental findings in the context of related literature. Various works have investigated AI energy consumption with the help of *CodeCarbon* [37, 36, 49, 14], however our findings indicate that their reported numbers are underestimations. The results also underline the importance of acknowledging the overhead impact of cooling [64, 32], which is one of the factors neglected by static and dynamic energy estimation approaches. It should be noted that holistic reporting on the environmental impacts of AI requires to investigate more factors of the AI life cycle [58], including intricate phenomena like embodied impacts [13] and rebound effects [35]. Nevertheless, I believe that reporting on the ground-truth energy consumption of running AI models remains a crucial ingredient for advancing the field in environmentally sustainable ways [15]. To that end, the investigations of this study can be useful for refining established tools, for example by incorporating constant factors that account for estimation errors or featuring explicit disclaimers about inaccuracies. From the experimental analysis, six important **takeaway points** can be formulated:

1. Energy consumption generally scales with model complexity and size, however certain AI architectures use their parameters more efficiently than others

2. Energy estimation approaches likely resulted in inaccurate reportings of environmental impacts, featuring errors that grow with model size and time

3. Static estimation hits a sweet spot with low errors for mid-sized models, but under- or overestimates the demand of large or small architectures by -40 to 40 %

4. Dynamic estimation better follows the ground-truth patterns, but consistently underestimates the consumption by 20–30%

5. Hyperparameters like batch size or temperature impact the efficiency and estimation errors, and accordingly should be carefully tuned and reported

6. Estimating the energy demand of CPU-only deployment results in errors that are considerably higher than for GPU inference

While providing important insights, this study also faces limitations, such as only evaluating a single environment. Other works have already evidenced that the choice of hardware and software impacts AI resource efficiency [17, 18], and this investigation should be seen as a first step toward more reliable energy estimations. The presented concepts and accompanying code repository allow to easily extend the investigation to other learning domains and evaluation setups. Future work could collect more data and deepen the analysis with regard to relations between configurations, environments, and resulting estimation errors. A second limitation lies in the external measurement approach, which might not be applicable to all AI deployment scenarios—this actually also holds true for static and dynamic estimations. For example, the energy consumption of compute clusters and data centers can be hardly monitored with basic energy meters, requiring different solutions for measuring ground-truth data, such as power distribution units with outlet level metering. Nevertheless, the proposed validation framework is applicable and affordable for on-premise deployment scenarios, which remains popular for the sake of privacy. What also remains for future work is a deeper investigation of how resource consumption trades with predictive quality [18] or other performance aspects [16], as past studies have focused on vision models [17]. In closing this discussion, I would like to emphasise once more that I fully support the aforementioned energy estimation initiatives—their published tools are invaluable resources for environmentally-aware practitioners, and I hope that my work will contribute to their continued improvement.

## 5 Conclusion

To summarize, this study validated energy estimation tools such as *CodeCarbon* and the *ML Impact Calculator*, which succeed at capturing the rough resource consumption of evaluated AI models. However, comparing their quantified numbers with externally measured ground-truth data also revealed significant estimation errors. Growing model complexity was observed to increase AI energy demand and estimation errors, which can even be further impacted by choice of hyperparameters and processor. With this work, resource-aware practitioners are equipped with means for investigating the actual energy consumption of their local AI experiments. Moreover, developers of energy estimation tools can improve their solutions along the guidelines of this work, accounting for estimation errors of their approaches. As such, the presented contributions establish transparency and not only promote but facilitate the sustainable development and use of AI.

## References


[1] Marah Abdin et al. *Phi-3 Technical Report: A Highly Capable Language Model Locally on Your Phone*. 2024. DOI: 10.48550/arXiv.2404.14219.

[2] Marah Abdin et al. *Phi-4 Technical Report*. 2024. DOI: 10.48550/arXiv.2412.08905.

[3] Yoshua Bengio et al. *International AI safety report*. Tech. rep. Accessed: 2025-09-12. AI Safety Institute, 2025. URL: https://coilink.org/20.500.12592/30e27x2.

[4] Samuel Bimenyimana and Godwin Asemota. "Traditional Vs Smart Electricity Metering Systems: A Brief Overview". In: *Journal of Marketing and Consumer Research* 46 (June 2018). URL: https://www.iiste.org/Journals/index.php/JMCR/article/view/42505.

[5] Abeba Birhane et al. "The Values Encoded in Machine Learning Research". In: *5th Conference on Fairness, Accountability and Transparency (FAccT)*. 2022, pp. 173–184. ISBN: 978-1-4503-9352-2. DOI: 10.1145/3531146.3533083.

[6] G. Bradski. "The OpenCV Library". In: *Dr. Dobb's Journal of Software Tools* (2000). URL: https://www.drdobbs.com/open-source/the-opencv-library/184404319.

[7] Michael Chiang and Jeffrey Morgan. *Ollama: Chat & build with open models*. Accessed: 2025-09-12. 2023. URL: https://ollama.com/.

[8] François Chollet et al. *Keras*. Accessed: 2025-09-12. 2015. URL: https://keras.io.

[9] François Chollet. “Xception: Deep Learning with Depthwise Separable Convolutions”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 1800–1807. DOI: 10.1109/CVPR.2017.195.

[10] Benoit Courty et al. *mlco2/codecarbon: v2.4.1*. May 2024. DOI: 10.5281/zenodo.11171501.

[11] DeepSeek-AI. *DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning*. 2025. DOI: 10.48550/arXiv.2501.12948.

[12] Jia Deng et al. “Imagenet: A Large-Scale Hierarchical Image Database”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2009, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

[13] Sophia Falk et al. *More than Carbon: Cradle-to-Grave environmental impacts of GenAI training on the Nvidia A100 GPU*. 2025. DOI: 10.48550/arXiv.2509.00093.

[14] Raphael Fischer. “Advancing the Sustainability of Machine Learning and Artificial Intelligence via Labeling and Meta-Learning”. PhD thesis. TU Dortmund University, 2025. DOI: 10.17877/DE290R-25716.

[15] Raphael Fischer, Thomas Liebig, and Katharina Morik. “Towards More Sustainable and Trustworthy Reporting in Machine Learning”. In: *Data Mining and Knowledge Discovery* (2024). ISSN: 1573-756X. DOI: 10.1007/s10618-024-01020-3.

[16] Raphael Fischer and Amal Saadallah. “AutoXPCR: Automated Multi-Objective Model Selection for Time Series Forecasting”. In: *Proceedings of the 30th International Conference on Knowledge Discovery and Data Mining (KDD)*. 2024, pp. 806–815. ISBN: 979-8-4007-0490-1. DOI: 10.1145/3637528.3672057. URL: https://doi.org/10.1145/3637528.3672057.

[17] Raphael Fischer et al. “A Unified Framework for Assessing Energy Efficiency of Machine Learning”. In: *Machine Learning and Principles and Practice of Knowledge Discovery in Databases*. 2022, pp. 39–54. DOI: 10.1007/978-3-031-23618-1_3.

[18] Raphael Fischer et al. “MetaQuRe: Meta-learning from Model Quality and Resource Consumption”. In: *Machine Learning and Knowledge Discovery in Databases*. 2024, pp. 209–226. ISBN: 978-3-031-70368-3. DOI: 10.1007/978-3-031-70368-3_13.

[19] Eva García-Martín et al. “Estimation of Energy Consumption in Machine Learning”. In: *Journal of Parallel and Distributed Computing* 134 (Dec. 2019), pp. 75–88. ISSN: 0743-7315. DOI: 10.1016/j.jpdc.2019.07.007.

[20] Gemma Team. *Gemma 3 Technical Report*. 2025. DOI: 10.48550/arXiv.2503.19786.

[21] Aaron Grattafiori et al. *The Llama 3 Herd of Models*. 2024. DOI: 10.48550/arXiv.2407.21783.

[22] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 770–778. DOI: 10.1109/CVPR.2016.90.

[23] Kaiming He et al. “Identity Mappings in Deep Residual Networks”. In: *Proceedings of the 14th European Conference on Computer Vision (ECCV)*. Ed. by Bastian Leibe et al. Vol. 9908. 2016, pp. 630–645. DOI: 10.1007/978-3-319-46493-0_38.

[24] Peter Henderson et al. “Towards the Systematic Reporting of the Energy and Carbon Footprints of Machine Learning”. In: *Journal of Machine Learning Research* 21 (248) (2020), pp. 1–43. DOI: 10.48550/arXiv.2002.05651.

[25] Miro Hodak, Masha Gorkovenko, and Ajay Dholakia. “Towards Power Efficiency in Deep Learning on Data Center Hardware”. In: *International Conference on Big Data*. 2019, pp. 1814–1820. DOI: 10.1109/BigData47090.2019.9005632.

[26] Andrew Howard et al. *Mobilenets: Efficient Convolutional Neural Networks for Mobile Vision Applications*. 2017. DOI: 10.48550/arXiv.1704.04861.

[27] Andrew Howard et al. “Searching for MobileNetV3”. In: *International Conference on Computer Vision (ICCV)*. Oct. 2019. DOI: 10.1109/ICCV.2019.00140.

[28] Gao Huang et al. “Densely Connected Convolutional Networks”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017, pp. 2261–2269. DOI: 10.1109/CVPR.2017.243.

[29] Intel Corporation. *Intel® 64 and IA-32 Architectures Software Developer’s Manual*. Accessed: 2025-09-08. 2024. URL: https://intel.com/content/www/us/en/developer/articles/technical/intel-sdm.html.

[30] Albert Q. Jiang et al. *Mistral 7B*. 2023. DOI: 10.48550/arXiv.2310.06825.

[31] Arpan Kumar Kar, Shweta Kumari Choudhary, and Vinay Kumar Singh. “How Can Artificial Intelligence Impact Sustainability: A Systematic Literature Review”. In: *Journal of Cleaner Production* (2022). DOI: 10.1016/j.jclepro.2022.134120.

[32] Imran Latif et al. *Cooling Matters: Benchmarking Large Language Models and Vision-Language Models on Liquid-Cooled Versus Air-Cooled H100 GPU Systems*. 2025. DOI: 10.48550/arXiv.2507.16781.

[33] Sebastian Lins et al. “Artificial Intelligence as a Service”. In: *Business & Information Systems Engineering* 63 (4) (Aug. 2021), pp. 441–456. ISSN: 1867-0202. DOI: 10.1007/s12599-021-00708-w.

[34] Zhuang Liu et al. “A ConvNet for the 2020s”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2022, pp. 11966–11976. DOI: 10.1109/CVPR52688.2022.01167.

[35] Alexandra Sasha Luccioni, Emma Strubell, and Kate Crawford. “From Efficiency Gains to Rebound Effects: The Problem of Jevons’ Paradox in AI’s Polarized Environmental Debate”. In: *8th Conference on Fairness, Accountability and Transparency (FAccT)*. 2025, pp. 76–88. ISBN: 9798400714825. DOI: 10.1145/3715275.3732007.

[36] Sasha Luccioni, Yacine Jernite, and Emma Strubell. “Power Hungry Processing: Watts Driving the Cost of AI Deployment?” In: *7th Conference on Fairness, Accountability and Transparency (FAccT)*. 2024, pp. 85–99. ISBN: 9798400704505. DOI: 10.1145/3630106.3658542.

[37] Sasha Luccioni, Sylvain Viguier, and Anne-Laure Ligozat. “Estimating the Carbon Footprint of BLOOM, a 176B Parameter Language Model”. In: *Journal of Machine Learning Research* 24 (253) (2023), pp. 1–15. URL: https://jmlr.org/papers/v24/23-0069.html.

[38] Sasha Luccioni et al. “Quantifying the Carbon Emissions of Machine Learning”. In: *NeurIPS Workshop on Tackling Climate Change with Machine Learning*. 2019. URL: https://climatechange.ai/papers/neurips2019/22.

[39] Jai Suphavadeeprasit Luigi Daniele. *Puffin dataset*. 2023. URL: https://huggingface.co/datasets/LDJnr/Puffin.

[40] Mistral-AI. *Magistral*. 2025. DOI: 10.48550/arXiv.2506.10910.

[41] NVIDIA Corporation. *NVIDIA Management Library (NVML)*. Accessed: 2025-09-08. 2012. URL: https://developer.nvidia.com/management-library-nvml.

[42] OpenAI. *gpt-oss-120b & gpt-oss-20b Model Card*. 2025. DOI: 10.48550/arXiv.2508.10925.

[43] Fabian Pedregosa et al. “Scikit-learn: Machine Learning in Python”. In: *Journal of Machine Learning Research (JMLR)* 12 (2011), pp. 2825–2830. DOI: 10.5555/1953048.2078195.

[44] Plotly Technologies Inc. *Plotly: Collaborative Data Science and Visualization*. Accessed: 2025-08-13. 2024. URL: https://plotly.com.

[45] Mark Sandler et al. “MobileNetV2: Inverted Residuals and Linear Bottlenecks”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 4510–4520. DOI: 10.1109/CVPR.2018.00474.

[46] Roy Schwartz et al. “Green AI”. In: *Communications of the ACM* 63 (12) (Nov. 2020), pp. 54–63. ISSN: 0001-0782. DOI: 10.1145/3381831.

[47] Jaime Sevilla et al. “Compute Trends Across Three Eras of Machine Learning”. In: *International Joint Conference on Neural Networks (IJCNN)*. 2022, pp. 1–8. DOI: 10.1109/IJCNN55064.2022.9891914.

[48] Karen Simonyan and Andrew Zisserman. “Very Deep Convolutional Networks for Large-Scale Image Recognition”. In: *3rd International Conference on Learning Representations (ICLR)*. Ed. by Yoshua Bengio and Yann LeCun. 2015. URL: https://arxiv.org/abs/1409.1556.

[49] Alexander van der Staay, Raphael Fischer, and Sebastian Buschjäger. “Stress-Testing USB Accelerators for Efficient Edge Inference”. In: *Proceedings of the 9th Symposium on Edge Computing (SEC)*. 2024, pp. 1–14. DOI: 10.1109/SEC62691.2024.00015.

[50] Emma Strubell, Ananya Ganesh, and Andrew McCallum. “Energy and Policy Considerations for Modern Deep Learning Research”. In: *AAAI Conference on Artificial Intelligence* 34 (09) (Apr. 2020), pp. 13693–13696. DOI: 10.1609/aaai.v34i09.7123.

[51] Christian Szegedy et al. “Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning”. In: *31st AAAI Conference on Artificial Intelligence*. 2017, pp. 4278–4284. DOI: 10.1609/AAAI.V31I1.11231.

[52] Christian Szegedy et al. “Rethinking the Inception Architecture for Computer Vision”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016, pp. 2818–2826. DOI: 10.1109/CVPR.2016.308.

[53] Mingxing Tan and Quoc Le. “EfficientNet: Rethinking Model Scaling for Convolutional Neural Networks”. In: *36th International Conference on Machine Learning (ICML)*. Ed. by Kamalika Chaudhuri and Ruslan Salakhutdinov. Vol. 97. Proceedings of Machine Learning Research. June 2019, pp. 6105–6114. URL: https://proceedings.mlr.press/v97/tan19a.html.

[54] Mingxing Tan and Quoc V. Le. “EfficientNetV2: Smaller Models and Faster Training”. In: *38th International Conference on Machine Learning (ICML)*. Ed. by Marina Meila and Tong Zhang. Vol. 139. 2021, pp. 10096–10106. URL: http://proceedings.mlr.press/v139/tan21a.html.

[55] Team OLMo. *2 OLMo 2 Furious*. 2025. DOI: 10.48550/arXiv.2501.00656.

[56] The pandas development team. *pandas-dev/pandas: Pandas*. Feb. 2020. DOI: 10.5281/zenodo.3509134.

[57] Gael Varoquaux, Sasha Luccioni, and Meredith Whittaker. “Hype, Sustainability, and the Price of the Bigger-is-Better Paradigm in AI”. In: *8th Conference on Fairness, Accountability and Transparency (FAccT)*. 2025, pp. 61–75. ISBN: 9798400714825. DOI: 10.1145/3715275.3732006.

[58] Carole-Jean Wu et al. *Sustainable AI: Environmental Implications, Challenges and Opportunities*. 2022. DOI: 10.48550/arXiv.2111.00364.

[59] Aimee van Wynsberghe. “Sustainable AI: AI for Sustainability and the Sustainability of AI”. In: *AI and Ethics* 1 (3) (Aug. 2021), pp. 213–218. ISSN: 2730-5961. DOI: 10.1007/s43681-021-00043-6.

[60] An Yang et al. *Qwen3 Technical Report*. 2025. DOI: 10.48550/arXiv.2505.09388.

[61] Matei A. Zaharia et al. “Accelerating the Machine Learning Lifecycle with MLflow”. In: *Data Engineering Bulletin* 41 (2018), pp. 39–45. URL: https://api.semanticscholar.org/CorpusID:83459546.

[62] Zeppelin GmbH. *Conversion Factors for $CO_2$ Emissions*. Accessed: 2025-09-12. 2024. URL: https://sustainabilityreport.zeppelin.com/2024/appendix/conversion-factors-co2-emissions/.

[63] Peiyuan Zhang et al. *TinyLlama: An Open-Source Small Language Model*. 2024. DOI: 10.48550/arXiv.2401.02385.

[64] Shuntao Zhu and Dan Wang. “Energy-efficient LLM Training in GPU datacenters with Immersion Cooling Systems”. In: *16th International Conference on Future and Sustainable Energy Systems*. 2025, pp. 407–414. ISBN: 9798400711251. DOI: 10.1145/3679240.3734609.

[65] Barret Zoph et al. “Learning Transferable Architectures for Scalable Image Recognition”. In: *Conference on Computer Vision and Pattern Recognition (CVPR)*. 2018, pp. 8697–8710. DOI: 10.1109/CVPR.2018.00907.